

A New Approach to PGP's Web of Trust

Rolf Haenni¹ and Jacek Jonczy²

¹ University of Applied Sciences Biel, CH-2501 Biel
Switzerland
rolf.haenni@bfh.ch

² University of Berne, CH-3012 Berne
Switzerland
jonczy@iam.unibe.ch

Abstract. Trust and authenticity networks are possible solutions for the key authenticity problem in a decentralized public-key infrastructure. A particular trust model, the so-called Web of Trust, has been proposed for and is implemented in the popular e-mail encryption software PGP and its open source derivatives like GnuPG. In this paper, we investigate the drawbacks and weaknesses of the current PGP trust model, and we propose a new approach to handle trust and key validity in a more sophisticated way. A prototype of our solution has been implemented and tested with a recent GnuPG release.³

1 Introduction

Public-key cryptography is a widely spread mechanism used for the establishment of secure communication. The exchanged encryption keys are *publicly* available, so that parties can securely communicate with each other over an insecure channel without having to agree upon a shared secret key beforehand. At first sight, this seems to be an ideal solution for the key exchange problem, but an important subproblem remains, namely to ensure that a public key actually belongs to its supposed owner. We will refer to it as the *key validation problem*. Note that spoofing another's identity is easily possible in any of several ways, i.e. the key validation problem is anything but trivial, particularly when the two users involved have never met and know nothing about each other.

As a solution for the key validation problem, *public-key infrastructures* (PKI) have been proposed and implemented in many different ways. One type of PKI requires one or several central authorities responsible for issuing *digital certificates* for public keys. Such a certificate is an unforgeable warranty for the binding between the involved public key and its owner.⁴ It is of crucial importance for the reliable operation of such a centralized PKI that the certificate authorities are fully *trustworthy*.

Another type of a PKI avoids central certificate authorities entirely. The most prominent example of such a decentralized PKI is a distributed trust model called *Web of Trust*. It is used in PGP, GnuPG, and other OpenPGP-compatible systems. The basic concept of this particular model goes back to Zimmermann's first PGP release in the early 90ies, and since then it has not changed much [12,15]. In this paper, we will refer to it as the *PGP trust model*, as suggested in [1].

Distributed trust models allow any user in the network to issue certificates for any other user.⁵ The issuers of such certificates are called *introducers*, who can make them publicly available, typically by uploading them to *key servers*, from which they are accessible to other users. Someone's personal collection of certificates is called *key ring*. In this way, responsibility for validating public keys is delegated to people you

³ This research is supported by the Swiss National Science Foundation, project no. PP002-102652.

⁴ Strictly speaking, a certificate is a warranty for the binding between the involved public key and a *description* of the owner [8]. Such a description can consist of a single attribute (name, first name, birth date, e-mail address, etc.) or a combination thereof. In the PGP context, this description is called *user ID* and typically consists of an e-mail address.

⁵ In the PGP jargon, issuing a certificate is called *signing a key*, and certificates are therefore called *signed public keys* or simply *signatures*.

trust. In comparison with a centralized PKI, this scheme is much more flexible and leaves trust decisions in the hands of individual users. These trust decisions are finally decisive for a user to validate public keys (i.e. to accept them as authentic) on the basis of the given local key ring.

In this paper, we will first give a short overview of the PGP trust model. The main goal is to point out some of its inherent weaknesses and deficiencies. To overcome these difficulties, we will then propose a more flexible PGP trust model, in which we propose to see the key validation problem as a two-terminal network reliability problem in a corresponding probabilistic graph [10]. This view is similar to the one proposed in [5], but it requires less theoretical background knowledge.

2 The PGP Trust Model

The PGP trust model has some particular characteristics. First of all, (only) three levels of trust are supported: *complete trust*, *marginal trust*, and *no trust*.⁶ The owner of the key ring, who needs to manually assign these trust values for all other users, automatically receives full trust (also called *implicit* or *ultimate* trust). When a user places trust in an introducer, implicitly it means that the user possesses a certain amount of confidence in the introducer's capability to issue valid certificates, i.e. correct bindings between users and public keys. This is the general intuition, but the actual meaning of the three trust levels in PGP is not clearly defined.

2.1 Key Validation in PGP

Based on such trust values, the PGP trust model suggests to accept a given public key in the key ring as *completely valid*, if either

- (a) the public key belongs to the owner of the key ring,
- (b) the key ring contains at least C certificates from completely trusted introducers with valid public keys,
- (c) the key ring contains at least M certificates from marginally trusted introducers with valid public keys.

To compensate for the above-mentioned ambiguity of the trust levels, the PGP trust model allows the users to individually adjust the two *skepticism parameters* C (also called `COMPLETES_NEEDED`) and M (also called `MARGINALS_NEEDED`). In general, higher numbers for these parameters imply that more people would be needed to conspire against you. The default values in PGP are $C = 1$ and $M = 2$, and $C = 1$ and $M = 3$ in GnuPG. If a given key is not completely valid according to the above rules, but if at least one certificate of a marginally or completely trusted introducer with a valid public key is present, then the key attains the status *marginally valid*. Otherwise, the key is considered to be *invalid*.⁷ The distinction between *marginally valid* and *invalid* keys is often neglected, so will we in the sequel.

Example. Consider the *certificate graph* shown in Fig. 1 which illustrates the PGP trust model. An arrow from X to Y represents a certificate issued by X for Y . Gray circles indicate complete trust (A, E, G, J, L, N), gray semicircles indicate marginal trust (C, D, F, M), and white circles indicate no trust (B, H, I, K, O). The results of the key validation are shown for $C = 1$ and $M = 2$. Completely valid public keys are represented by nested circles ($A, B, C, D, E, H, I, J, N, O$). Note that all public keys with a certificate issued by A , the owner of the key ring, are completely valid.

⁶ It is common to separate *unknown trust* from *no trust*, but this has no significance for the PGP key validation algorithm.

⁷ A more general way of defining the validity of public keys is by means of the so-called *key legitimacy* $L = c/C + m/M$, where c and m denote the number of certificates from completely resp. marginally trusted introducers with valid keys [13]. Then a key is completely valid for $L \geq 1$, marginally valid for $0 < L < 1$, and invalid for $L = 0$.

2.2 How Trustworthy is the PGP Trust Model?

The PGP trust model has both advantages and drawbacks. An important advantage is the simplicity of the above-mentioned evaluation rules. This leads to a very efficient evaluation algorithm, which performs on a given key ring in time linear to its size. Another advantage is the above-mentioned adjustability of the skepticism parameters C and M , which allows the users to express their own policy regarding the threshold of his confidence in the PGP key validation mechanism.⁸

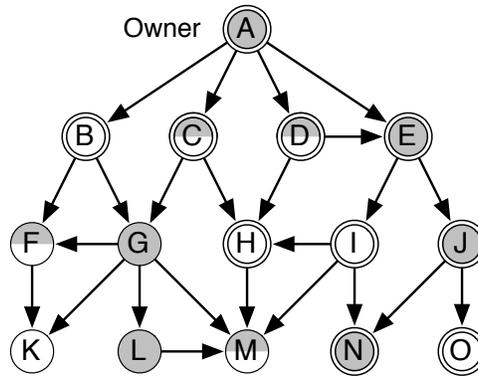


Fig. 1. Example of a PGP key ring owned by user A.

Limited trust levels. A major deficiency of the PGP trust model is that the limited levels of trust in PGP is clearly insufficient to reflect possible varying opinions about an introducer's trustworthiness.⁹ In real life, it may well be that among two marginally trustworthy introducers one of them is twice more trustworthy than the other. Unfortunately, the PGP trust model does not support such distinctions.

Limited validity levels. A similar problem arises from the limited levels of validity, which does not always allow to properly separate quite different situations within a certificate graph. As an example, consider two different keys with an unequal number of certificates, all from marginally trusted introducers. If in both cases the number of certificates is beyond the threshold M , according to the third key validation rule, the keys will be rated equally as *completely valid*. This conclusion does not measure up to the fact that more positive evidence is available for the validity of the key with the greater number of certificates.

Counter-intuitive key validation. Another quite severe problem arises from the pragmatic nature of the third key validation rule. As demonstrated in [5,9], this can lead to quite counter-intuitive conclusions. Consider the two key rings shown in Fig. 2. On the left hand side, there are two certificate chains of length 3 from A to B , each of them containing one completely trusted and one marginally trusted introducer. On the right hand side, the situation is very similar, except that there are more than two (possibly infinitely many) certificate chains of length 3 from A to B , in which the order of the two introducers is reversed. In such a situation, one would clearly expect B to have a higher degree of validity in the second case, but the PGP trust model tells us to accept B in the first and reject B in the second case (for the default values $C = 1$ and $M = 2$).

⁸ Another adjustable PGP parameter is CERT_DEPTH, which defines the maximum certification chain length. As in the example of the previous subsection, this parameter is often ignored.

⁹ It is interesting to know that the OpenPGP message format specification reserves an entire octet to store trust values: "The trust amount is in a range from 0–255, interpreted such that values less than 120 indicate partial trust and values of 120 or greater indicate complete trust. Implementations should emit values of 60 for partial trust and 120 for complete trust" [3].

Hidden key dependencies. A similar type of problem arises from the fact that the PGP trust model does not take into account the possibility of people controlling multiple public keys. This results from the fact that trust is actually assigned to keys, and not to users. As a consequence, it could well happen that a key with certificates from two marginally trusted and apparently different users is considered to be valid (for $M = 2$), but in reality they are issued by the one and the same person. This is a problem of invisible dependencies, which could easily be exploited by malicious users to make people accept non-existing bindings between users and keys [9].

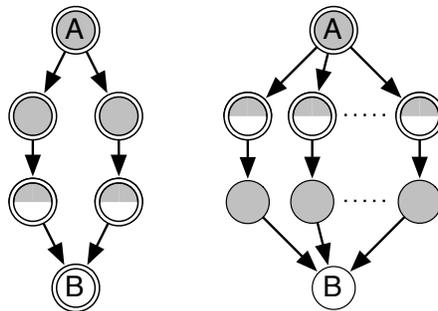


Fig. 2. Counter-intuitive PGP key validation.

To conclude this subsection, let us mention two features of more sophisticated trust models, which are not available in the PGP trust model. The first one is the ability to issue *recommendations* (i.e. signed statements relative to somebody’s trustworthiness as a reliable introducer) or other higher level statements in the sense of Maurer’s multi-level trust model [11].¹⁰ The second missing feature is the support of negative, mixed, or weighted statements as proposed in [6].

3 Probabilistic Key Validation

To overcome some of the above-mentioned deficiencies of the PGP trust model, we propose to translate the key validation problem into an appropriate *network reliability problem* [4]. Network reliability problems are well-studied in *reliability theory*, and they have many applications in network design and other areas.

In a general setting, the starting point is a network represented as a directed probabilistic graph, where vertices are subject to independent failures with given probabilities, and arcs (directed edges) are perfectly reliable. The problem then is to compute the probability that the network provides an operating connection between two, some, or all vertices. For the particular application of this paper, only the so-called *source-to-terminal* or *s,t-connectedness problem* is relevant. This is the problem of computing the probability of establishing at least one operating network path from a vertex s (the source) to another vertex t (the terminal).

3.1 Formulating Trust-Based Key Validation as a Network Reliability Problem

If we intuitively map the *s,t-connectedness* problem to the trust-based key validation context, we get the following setting: the graph represents the key ring, vertices are introducers (resp. their public keys), arcs

¹⁰ The OpenPGP message format specification foresees the possible inclusion of higher level certificates such as recommendations: “Signer asserts that the key is not only valid, but also trustworthy, at the specified level. Level 0 has the same meaning as an ordinary validity signature. Level 1 means that the signed key is asserted to be a valid trusted introducer, [...]. Level 2 means that the signed key is asserted to be trusted to issue level 1 trust signatures, i.e. that it is a “meta introducer”. Generally, a level n trust signature asserts that a key is trusted to issue level $n-1$ trust signatures” [3].

are certificates, and vertex reliabilities are trust values assigned to introducers. In other words, a trust value is now understood as somebody's *probability* of being a reliable introducer.¹¹ This allows us to specify trust values on an infinitely fine scale between 0 (no trust) and 1 (complete trust).

The example in Fig. 3 depicts the subgraph obtained from the key ring of Fig. 1, if one is concerned with the validity of K 's public key only. The PGP trust values are replaced by respective probabilities: 0.9 for complete trust, 0.5 and 0.6 for marginal trust, and 0.1 for no trust. Note that the trust value assigned to K has no impact on its own key validation.

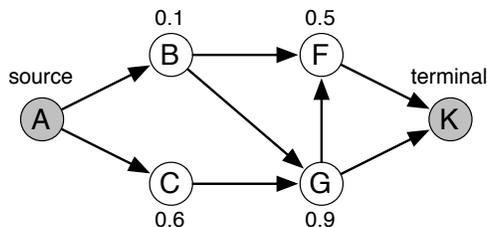


Fig. 3. A PGP key ring as reliability network.

The translation into a network reliability problem offers us now a broad range of computational techniques to solve the key validation problem. We will first use the above example to illustrate a simple but not very efficient method, and then give a short overview of more advanced techniques.

3.2 Computing Key Validity

Consider again the network of Fig. 3, in which the key validation problem consists in A 's attempt to validate K 's public key. This means to compute the s,t-connectedness with A as source and K as terminal. For this, there must be at least one complete certificate path from A to K , and if no such path exists, the validity of K 's public key cannot be established from A 's point of view. In the example of Fig. 3, there are five certificate paths from A to K , namely $\{A, B, F, K\}$, $\{A, B, G, K\}$, $\{A, C, G, K\}$, $\{A, C, G, F, K\}$, and $\{A, B, G, F, K\}$. Note that the last two paths are not minimal and therefore irrelevant for the overall network reliability. From the remaining three paths, we can also omit the source A (who's trust value is 1 by default) and the terminal K (who's trust value has no impact).¹² Finally, we obtain the following set of minimal paths:

$$\text{minpaths}_{A,K} = \{\{B, F\}, \{B, G\}, \{C, G\}\}.$$

To calculate the network reliability for a connection from A to K , which will be our measure for the validity of K 's public key, we have to compute the probability of the set $\text{minpaths}_{A,K}$. The probability of a single path is simply the product of its (stochastically independent) trust values, and for the overall probability of the set $\text{minpaths}_{A,K}$, we can apply the so-called *inclusion-exclusion* formula:

$$\begin{aligned} P(\text{minpaths}_{A,K}) &= P(\{B, F\}) + P(\{B, G\}) + P(\{C, G\}) \\ &\quad - P(\{B, F, G\}) - P(\{B, F, C, G\}) - P(\{B, C, G\}) \\ &\quad + P(\{B, F, C, G\}) = 0.581. \end{aligned}$$

This result is the probabilistic measure we propose for the validity of K 's public key. Depending on A 's own validation policy, e.g. by specifying a validity threshold $\lambda \in [0, 1]$, the key may be accepted as valid or not. For instance, if A has a strict acceptance policy, she sets accordingly a high threshold, say $\lambda = 0.9$. In this case, A would not accept K 's public key as valid, since $0.581 < \lambda$. On the other hand, A would

¹¹ We do not specify whether these probabilities are interpreted as frequencies or as subjective degrees of belief.

¹² This is a slight deviation from the classical s,t-connectedness problem, where every vertex in the path is relevant, including the source and the terminal.

neither reject the key, but rather collect more evidence in form of further certificates. Such an evaluation is very different from the PGP scenario in Fig. 1, where K 's public key is considered simply invalid (except for $C = 1$ and $M = 1$).

For the solution of the s,t-connectedness (and other network reliability problems), many alternative and more sophisticated techniques exist. However, we will not further go into this topic and refer for more information to the corresponding literature [4,2].

Our proposal for a probabilistic evaluation of trust networks solves some of the deficiencies of the PGP trust model mentioned in Section 2. First of all, it eliminates the limited levels of trust and validity, which leads to an increased overall flexibility. At the same time, it solves the problem of counter-intuitive conclusions in situations like the one shown in Fig. 2. This improves both the robustness and the coherence of the results. Furthermore, the method has been implemented in GnuPG release v.1.4.5, and the probabilistic measure is computed there by means of a Monte-Carlo sampling algorithm. The implementation is discussed in [7], and in more detail in [14].

4 Conclusion

The main contribution of this paper is the proposal for a probabilistic trust model for PGP and its derivatives. The key validation problem has been transformed into a source-to-terminal network reliability problem. As a result, several weaknesses of PGP's trust model are eliminated. The most important improvement comes from the gradual trust values, which then result in gradual levels of validity. Our new model also avoids some counter-intuitive scenarios. To conclude, we think that the proposed key validation method is a reasonable, flexible, and useful enhancement of the existing GnuPG functionality. At the moment, it is unfortunately not officially included yet in the GnuPG software.

References

1. A. Abdul-Rahman. The PGP trust model. *EDI-Forum: the Journal of Electronic Commerce*, 10(3):27–31, 1997.
2. M. O. Ball, C. J. Colbourn, and J. S. Provan. Network reliability. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 673–762. Elsevier, 1995.
3. J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. *RFC 2440: OpenPGP Message Format*. IETF Network Working Group, 1998.
4. C. J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, New York, USA, 1987.
5. R. Haenni. Using probabilistic argumentation for key validation in public-key cryptography. *International Journal of Approximate Reasoning*, 38(3):355–376, 2005.
6. J. Jonczyk and R. Haenni. Credential networks: a general model for distributed trust and authenticity management. In A. Ghorbani and S. Marsh, editors, *PST'05, 3rd Annual Conference on Privacy, Security and Trust*, pages 101–112, St. Andrews, Canada, 2005.
7. J. Jonczyk, M. Wüthrich, and R. Haenni. A probabilistic trust model for GnuPG. In *23C3, 23rd Chaos Communication Congress*, pages 61–66, Berlin, Germany, 2006.
8. R. Kohlas, J. Jonczyk, and R. Haenni. Towards precise semantics for authenticity and trust. In *PST'06, 4th Annual Conference on Privacy, Security and Trust*, pages 124–134, Toronto, Canada, 2006.
9. R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In H. Imai and Y. Zheng, editors, *PKC'2000, Third International Workshop on Practice and Theory in Public Key Cryptography*, LNCS 1751, pages 93–112, Melbourne, Australia, 2000. Springer.
10. G. Mahoney, W. Myrvold, and G. C. Shoja. Generic reliability trust model. In A. Ghorbani and S. Marsh, editors, *PST'05: 3rd Annual Conference on Privacy, Security and Trust*, pages 113–120, St. Andrews, Canada, 2005.
11. U. Maurer. Modelling a public-key infrastructure. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *ESORICS, European Symposium on Research in Computer Security*, LNCS 1146, pages 324–350. Springer, 1996.
12. W. Stallings. *Protect Your Privacy, a Guide for PGP Users*. Prentice Hall, 1995.
13. W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 3rd edition, 2003.
14. M. Wüthrich. GnuPG and probabilistic key validation. Bachelor thesis, IAM, University of Berne, Switzerland, 2006.
15. P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1994.